Improving Recognition of Novel Input with Similarity

Jerod J. Weinman and Erik Learned-Miller
{weinman,elm}@cs.umass.edu
Department of Computer Science
University of Massachusetts-Amherst
Amherst, MA 01002

Abstract

Many sources of information relevant to computer vision and machine learning tasks are often underused. One example is the similarity between the elements from a novel source, such as a speaker, writer, or printed font. By comparing instances emitted by a source, we help ensure that similar instances are given the same label. Previous approaches have clustered instances prior to recognition. We propose a probabilistic framework that unifies similarity with prior identity and contextual information. By fusing information sources in a single model, we eliminate unrecoverable errors that result from processing the information in separate stages and improve overall accuracy. The framework also naturally integrates dissimilarity information, which has previously been ignored. We demonstrate with an application in printed character recognition from images of signs in natural scenes.

1. Introduction

The problem of character recognition in document analysis has a long history and is one of the most successful applications of computer vision, image processing, and machine learning techniques. However, faced with complications such as noisy input, novel fonts, and unconstrained text in natural images, the performance of traditional OCR systems degrades more rapidly than humans' ability to read the same text. A possible reason for this could be that people are able to apply many more sources of information to the problem than current automated techniques. This is not unique to character recognition, of course; using more information sources in our approaches to many computer vision problems should improve our results. In this work, we integrate appearance similarity, one underused source of information, in a unified probabilistic framework to reduce false matches by a factor of four and improve overall accuracy.

Progress has been made recently in the task of automatically detecting and reading relatively small amounts of printed text (e.g., signs) from natural images [5, 6] as an aid to the visually impaired or travellers in need of translation. While the fundamental task of character recognition is the same as in traditional document analysis, there are some important differences that can drastically affect performance. Perspective projection from non-uniform imaging conditions can alter the appearance of characters requiring rectification before recognition [5]. Signs are also typically printed in a wider variety of fonts than average documents, due to glyph alterations and custom designs. Finally, the number of characters in a given sign is relatively small, while the amount of text in a document can be quite large.

Recent advances in OCR performance have exploited the length of documents. Hong and Hull [11] cluster word images and then label the clusters. Similarly, Breuel learns a probability of whether two images contain the same character and uses the probability to cluster individual characters [2], with subsequent cluster labeling (i.e., by voting) and nearest neighbor (most similar) classification [3]. These methods capitalize on the idea of similarity; that characters and words of similar appearance should be given the same label. However, they suffer from the drawback that there is no feedback between the labeling and clustering process. Hobby and Ho [10] ameliorate this somewhat by purging outliers from a cluster and matching them to other clusters where possible. These processes all solve the clustering and recognition problems in separate stages, making it impossible to recover from errors in the clustering stage.

Thus far, the *dissimilarity* between character images has not been used as evidence *against* giving them the same label, but in many circumstances this is a reasonable approach. (If there are multiple fonts present, then the font identity may also be considered part of the label.) The previous clustering-based methods only ensure that all cluster members are given the same label; they do not prevent different clusters from being assigned the same label.

Consider the example in Figure 1. The top row of text is

To appear in Andrew Fitzgibbon, Yann LeCun, and Camillo J. Taylor, editors, *IEEE Conference on Computer Vision and Pattern Recognition*, New York, June 2006.

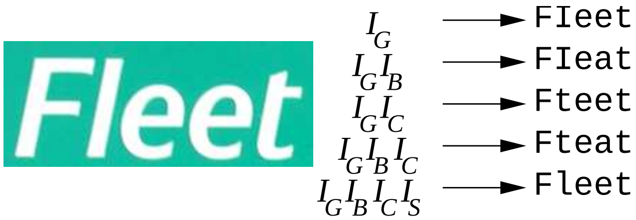

Figure 1. A query image (left) is interpreted with varying amounts of image (I_G) and linguistic (I_B, I_C) information. Only with similarity information (I_S) is the other contextual information constrained to global consistency.

the result of reading the sign on the left using only basic information about character images, and the lowercase `l` (ell) is mistaken for an uppercase `I` (eye). The next two results each combine the image information with some basic language information. These do not correct the error but in fact introduce new errors. Combining these three information sources in the fourth line elicits both new errors. The image and language information is based on local context and does not require any global consistency. By adding similarity information in the last line, the errors are corrected; the two `e` characters that appear the same are given the same label, while the `l` and `t` characters of dissimilar appearance are given different labels.

Our recognition strategy improves on two issues lacking in previous approaches. First, by simultaneously incorporating character identity and similarity information into a unified probabilistic model, we eliminate the need for distinct clustering/recognition steps and the potential for unrecoverable errors. Second, we treat similarity and dissimilarity as two sides of the same coin, which prevents dissimilar characters from being given the same label. The rest of the paper presents our probabilistic framework, including available information and related features, followed by a set of experiments on reading text in images of real signs. We then discuss the results and conclude that a unified method including similarity information significantly improves accuracy and reduces false matches by fourfold.

2. Probabilistic Framework for Recognition

Graphical models of probability are a powerful tool for describing and modeling the logical dependence of various information sources and unknowns in a Bayesian framework. We employ a discriminative undirected graphical model [13] for predicting character identities.

Let $\mathbf{x}$ be an input image representation and $\mathbf{y}$ the string of characters contained in the image, taken from an alphabet $\mathcal{A}$. Letting I represent our information about the problem, including any assumptions that give rise to the choice of a particular probability, we frame the task of reading text in images as an inference problem—using I and some training data $\mathcal{D}$—over a model or parameter space Θ:

$$p(\mathbf{y} \mid \mathbf{x}, \mathcal{D}, I) = \int_{\boldsymbol{\Theta}} p(\mathbf{y} \mid \mathbf{x}, \boldsymbol{\theta}, I)\, p(\boldsymbol{\theta} \mid \mathcal{D}, I)\, \mathrm{d}\boldsymbol{\theta}. \quad (1)$$

Note we have assumed that (i) given a prediction model $\boldsymbol{\theta}$, the training data $\mathcal{D}$ do not reveal anything additional about $\mathbf{y}$, and (ii) given the training data $\mathcal{D}$, an additional image $\mathbf{x}$ does not give any information about the prediction model $\boldsymbol{\theta}$. Of course, evaluating such an integral is non-trivial, so we take the standard approach of finding the most likely model

$$\widehat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} p(\boldsymbol{\theta} \mid \mathcal{D}, I) \quad (2)$$

and using the point approximation

$$p(\boldsymbol{\theta} \mid \mathcal{D}, I) = \delta\left(\boldsymbol{\theta} - \widehat{\boldsymbol{\theta}}\right) \quad (3)$$

so that the integral (1) becomes

$$p(\mathbf{y} \mid \mathbf{x}, \mathcal{D}, I) \approx p\left(\mathbf{y} \mid \mathbf{x}, \widehat{\boldsymbol{\theta}}, I\right). \quad (4)$$

The probability $p(\mathbf{y} \mid \mathbf{x}, \boldsymbol{\theta}, I)$ is the typical undirected graphical model: the unknown characters $\mathbf{y}$ are indexed by the nodes of a graph, and an edge indicates logical dependence between two nodes. Such a model may be written

$$p(\mathbf{y} \mid \mathbf{x}, \boldsymbol{\theta}, I) = \exp\left\{-U(\mathbf{y}, \mathbf{x}; \boldsymbol{\theta}) - \log Z(\mathbf{x}; \boldsymbol{\theta})\right\}, \quad (5)$$

where $U(\mathbf{y}, \mathbf{x}; \boldsymbol{\theta})$ is an energy function for observations and predictions parameterized by $\boldsymbol{\theta}$. The normalizing value

$$Z(\mathbf{x}; \boldsymbol{\theta}) = \sum_{\mathbf{y}} \exp\left\{-U(\mathbf{y}, \mathbf{x}; \boldsymbol{\theta})\right\} \quad (6)$$

ensures that (5) is a proper probability. Considering up to pairwise dependencies, the energy is decomposed into local and pairwise terms

$$\begin{aligned}
U(\mathbf{y}, \mathbf{x}; \boldsymbol{\theta}) &= U_L(\mathbf{y}, \mathbf{x}; \boldsymbol{\theta}) + U_P(\mathbf{y}, \mathbf{x}; \boldsymbol{\theta}) && (7)\\
U_L(\mathbf{y}, \mathbf{x}; \boldsymbol{\theta}) &= \sum_i U_i(y_i, \mathbf{x}; \boldsymbol{\theta}) && (8)\\
U_P(\mathbf{y}, \mathbf{x}; \boldsymbol{\theta}) &= \sum_{(i,j)} U_{ij}(y_i, y_j; \boldsymbol{\theta}) + \\
&\quad \sum_{(i,j)} U_{ij}(y_i, y_j, \mathbf{x}; \boldsymbol{\theta}) && (9)
\end{aligned}$$

where U_L is a sum of local energies for a character and given image, and U_P is a sum of pairwise energies for two character labels, which may or may not also depend on the image. Incorporating the pairwise, image-dependent terms will allow for image similarity comparisons when predicting labels.

The factor graph [12] of the final model incorporating all the information, which corresponds to the probability

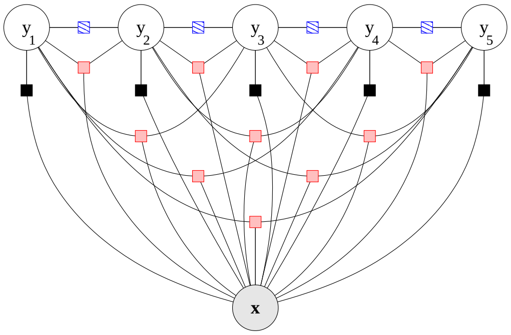

Figure 2. Factor graph for inferring characters $\mathbf{y}$ from a given image $\mathbf{x}$. The solid (black) factors capture relationships between the image and character identity. Hatched (blue) factors between neighboring y capture language information including bigrams and letter case. Shaded (red) factors among $\mathbf{y}$ account for similarities between characters in $\mathbf{x}$ for jointly labeling the string.

(5), is shown in Figure 2. Each term of the total energy (7) belongs to one factor. Factor graphs are similar to the usual graphical model representation, but the actual clique parameterization is represented. For instance, the equivalent graphical model for our probability is fully connected among the y_i, which would allow a single energy term to depend on all of $\mathbf{y}$. The factor graph illustrates that only pairwise terms are being used. See Kschischang et al. [12] for further background on the factor graph representation of probability models.

Given data $\mathcal{D} = \left\{\mathbf{y}^{(k)}, \mathbf{x}^{(k)}\right\}_k$ consisting of a sequence of labeled observations, the optimization (2) is the usual maximum *a posteriori* (MAP) estimation with some parameter prior $p\left(\boldsymbol{\theta} \mid I\right)$ [13]. Using Bayes' rule, the parameter posterior is

$$p\left(\boldsymbol{\theta} \mid \mathcal{D}, I\right) \propto p\left(\boldsymbol{\theta} \mid I\right) \prod_k p\left(\mathbf{y}^{(k)} \mid \mathbf{x}^{(k)}, \boldsymbol{\theta}, I\right) \tag{10}$$

where the product terms have the same model form (5). After taking logarithms, the objective function is given by

$$\begin{aligned}\mathcal{L}\left(\boldsymbol{\theta}; \mathcal{D}\right) &= \sum_k \left[-U\left(\mathbf{y}^{(k)}, \mathbf{x}^{(k)}; \boldsymbol{\theta}\right)\right. \\ &\quad \left. - \log Z\left(\mathbf{x}^{(k)}; \boldsymbol{\theta}\right)\right] + \log p\left(\boldsymbol{\theta} \mid I\right).\end{aligned} \tag{11}$$

When U is linear in $\boldsymbol{\theta}$, the objective (11) is convex (assuming the parameter prior is convex), so $\widehat{\boldsymbol{\theta}}$ can be found by convex optimization. The inference task for the model (5) will involve calculating the normalizing partition function (6), marginal probabilities for the y_i, or the $\mathbf{y}$ of maximum posterior probability. While inference is efficient when the corresponding factor graph is cycle-free, it is intractable in general. When the factor graph has cycles, we must resort to approximate inference techniques for calculating $\log Z$ and the gradient of the objective $\mathcal{L}$, in which case the optimization of (11) is only approximate.

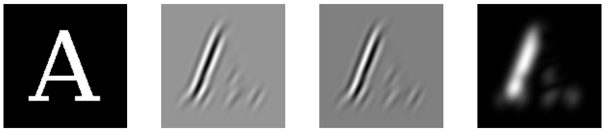

Figure 3. An example training character with (left to right) real, imaginary, and complex modulus filter responses for one orientation and scale.

The information we apply to this task consists broadly of two types: visual and linguistic. Previous work has shown text detection can be done fairly reliably [6, 18], effectively performing affine rectification and giving character bounding boxes [5]. In this work, we take the locations of characters as a given with the image, letting $\mathbf{x}_i$ denote the features of an image patch relevant to the ith character y_i. For now, we assume that each observation $\mathbf{x}$, (representing one local region of text) contains a single font. Our model's parameters $\boldsymbol{\theta} = \left(\mathbf{w}, \mathbf{b}, \mathbf{l}, \mathbf{s}\right)$ are partitioned by the information they incorporate, and we estimate each subset independently. Although they are not independent in general, our results show this to be an acceptable approximation. Next, we describe the information and features used in our model.

2.1. Character Image Features

Gabor filters are an effective and widely used tool for feature extraction that decompose geometry into local orientation and scale [7]. Their success in handwriting recognition [8] and printed character recognition [5] demonstrates their utility for this task. Using a minimally redundant design strategy [14], a bank of 18 filters spanning three scales (5, 10, and 20 pixels/cycle) and six orientations (30° increments from 0° to 150°) is applied to the grayscale image, yielding complex coefficients that contain phase information (the real and imaginary parts of the filter are even and odd functions, respectively). Taking the complex modulus of the filter outputs provides phase invariance and makes the responses less sensitive to translations of the input; see Figure 3. Henceforth, let $\mathbf{x}$ denote a vector of these Gabor filter responses to the original input image, with $|\mathbf{x}|$ the vector of component-wise complex moduli of the responses. Similarly, the entries of $\mathbf{x}$ corresponding to the ith character are given by the vector $\mathbf{x}_i$, with moduli $|\mathbf{x}_i|$.

Let us assume that there is a relationship between the identity of the character and the filter responses, which signify local scale and orientation; this information is denoted I_G for the Gabor decomposition. We then associate character classes with these filtered images by a linear energy

$$U_i^G\left(y_i, \mathbf{x}; \mathbf{w}\right) = \mathbf{w}\left(y_i\right)^\top \left|\mathbf{x}_i\right|, \tag{12}$$

where $\mathbf{w}\left(c\right)$ is a real-valued vector of weights for a particular character c. The weights $\mathbf{w} = (\mathbf{w}(c))_c$ are optimized as described earlier by maximizing $p(\mathbf{w} \mid \mathcal{D}, I_G)$ for $\mathbf{w}$, where $\mathcal{D}$ is a sequence of image/character pairs that are independent given the model $\mathbf{w}$. We use a Laplacian prior [9, 17]

$$p(\mathbf{w} \mid \alpha, I_G) \propto \exp\{-\alpha \|\mathbf{w}\|_1\}, \tag{13}$$

where $\|\mathbf{w}\|_1$ is the ℓ_1 vector norm; α is chosen by validation. Experimental details may be found in Section 3.2.

2.2. Language Features

Properties of the language are strong cues for recognizing characters in previously unseen fonts and under adverse conditions. We add these to the model in the form of two information sources: character bigrams and letter case.

It is well known that the English lexicon employs certain character juxtapositions more often than others. N-grams are a widely-used general feature for character and handwriting recognition [1]. Our model uses this information I_B via the linear features

$$U_{ij}^{B}(y_i, y_j; \mathbf{b}) = b(y_i, y_j) \tag{14}$$

where i and j are ordered, adjacent characters, and $\mathbf{b} = (b(c, c'))_{c,c'}$ are real-valued weights for each bigram cc'. We do not consider letter case to be important in bigrams, so the weights in $\mathbf{b}$ are tied across case (i.e., $b(\mathtt{R}, \mathtt{A}) = b(\mathtt{r}, \mathtt{A}) = b(\mathtt{R}, \mathtt{a}) = b(\mathtt{r}, \mathtt{a})$). These weights are also optimized independently by maximizing $p(\mathbf{b} \mid \mathcal{T}, I_B)$ for $\mathbf{b}$, where $\mathcal{T}$ is a corpus of English text. We use a uniform prior $p(\mathbf{b} \mid I_B) \propto 1$.

Prior knowledge of letter case with respect to words also proves important for accurate recognition. In some fonts potentially confusable characters may have different cases (e.g., `l` and `I`, lowercase ell and capital eye). Since we do not binarize the images, there is no direct method for measuring the relative size of neighboring characters. We can improve recognition accuracy in context because English rarely switches case in the middle of the word. Additionally, uppercase to lowercase transitions are common at the beginning of words, but the reverse is not. (Note that digit characters have no case.) This information I_C is incorporated with the feature weights

$$U_{ij}^{C}(y_i, y_j; \mathbf{l}) = \begin{cases} l_s & y_i, y_j \text{ same case} \\ l_d & y_i, y_j \text{ different case} \\ 0 & \text{otherwise} \end{cases} \tag{15}$$

when i and j are adjacent characters within a word and

$$U_{ij}^{C}(y_i, y_j; \mathbf{l}) = \begin{cases} l_u & y_i \text{ lowercase, } y_j \text{ uppercase} \\ 0 & \text{otherwise} \end{cases} \tag{16}$$

when i and j are the first and second characters of a word, respectively. We set the parameters $\mathbf{l} = \begin{bmatrix} l_s & l_d & l_u \end{bmatrix}^\top$ by maximizing $p(\mathbf{l} \mid \mathcal{T}, I_C)$ with the same corpus used for the bigram features. A uniform prior is also used for $\mathbf{l}$.

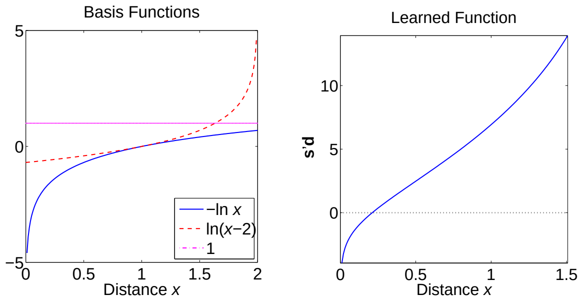

Figure 4. Similarity basis functions and the learned energy for the distance between different images of the same character; the coefficients are $\mathbf{s} = \begin{bmatrix} 0.9728 & 9.3191 & 6.9280 \end{bmatrix}$. The dotted line in the right-hand figure shows the crossover from reward to penalty, which occurs at an angle of about 37°.

2.3. Similarity

An important, underused source of information for recognition is the similarity among the character images themselves—two character images that look the same should rarely be given different labels. Toward this end, we need a comparison function for images. We have found the vector angle between the concatenated real and imaginary parts of filtered image vectors $\mathbf{x}_i$ and $\mathbf{x}_j$ to be a robust indicator of image discrepancies. Letting θ be the angle between two such vectors, we use $d = 1 - \cos\theta$ as a distance measure, which has range $[0, 2]$. When the distance is small the characters are very similar, but when large they are dissimilar. Using this information I_S we add the features

$$U_{ij}^{S}(y_i, y_j, \mathbf{x}; \mathbf{s}) = \delta(y_i, y_j)\, \mathbf{s}^\top \mathbf{d}_{ij} \tag{17}$$

where $\delta(c, c')$ is the Kronecker delta, and

$$\mathbf{d}_{ij} = \begin{bmatrix} \ln(d_{ij}) & -\ln(2 - d_{ij}) & 1 \end{bmatrix}^\top \tag{18}$$

is a vector of basis functions that transform the distance d_{ij} between two character images $\mathbf{x}_i$ and $\mathbf{x}_j$. The first two functions each have a distance range boundary as an asymptote, and the last is a bias term. Thus, the first weight in $\mathbf{s}$ establishes a low energy reward for small distances, the second weight a high energy penalty for larger distances, and the bias helps (in conjunction with the first two) establish the crossover point. This is similar to the inverse of the sigmoid function with a scaled range, except that it is no longer symmetric about the zero-crossing; see Figure 4. These weights are again optimized independently by maximizing $p(\mathbf{s} \mid \mathcal{S}, I_S)$, where $\mathcal{S}$ contains paired character images that are the same up to small affine transformations and noise, as well as pairs of different character images. See Section 3.1 for more details. We also use a uniform prior for $\mathbf{s}$.

Since all previous features were either local to characters or formed a chain along the text, inference in models including I_G, I_B, or I_C is fast and exact via the sum-product algorithm (belief propagation) [12]. However, by introducing I_S we are now making pairwise comparisons between *all* characters of the query $\mathbf{x}$. Inference in such a cyclic graph becomes intractable and will require approximate solutions. We use a loopy sum-product approximation and encounter few problems with convergence.

3. Experiments

In this section we describe the results of recognizing text in images of real signs and provide details on the data and procedure used to train our model. Our alphabet of characters A consists of 26 lowercase, 26 uppercase, and the 10 digit characters (62 total).

3.1. Data Sets

We generated images of each character in 1,000 commercially available fonts using GIMP.[1] Each image is 128×128 pixels with the font height at 100 pixels; the bounding box of the character is centered in the window.

A corpus of English text was acquired from Project Gutenberg[2]—84 books including novels, non-fiction, and reference for a total of more than 11 million words and 49 million characters (from our 62 character alphabet).

For learning the similarity function (17), we generated pairs of the same character (in the same font) and pairs of different characters (also in the same font) with the following procedure. First, we select a font and a character uniformly at random. To produce a similar character, we generate a random linear transformation

$$T = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} \sigma_x & \rho_x \\ \rho_y & \sigma_y \end{bmatrix} \qquad (19)$$

with rotation $\theta \sim \mathcal{N}(0, 1^\circ)$, scale factors $\sigma_x, \sigma_y \sim \mathcal{N}(1, 0.01)$, and skew factors $\rho_x, \rho_y \sim \mathcal{N}(0, 0.005)$. This transformation is applied to the original image, followed by additive noise. To produce a dissimilar pair, a different character is chosen uniformly at random. We choose different characters from the same font because these are likely to be more similar than different characters from different fonts, allowing for a better threshold to be learned. Additive Gaussian noise $\epsilon \sim \mathcal{N}(0, 0.01)$ is added to the original and transformed images prior to Gabor filtering. For optimal predictive discrimination, the ratio of same to different pairs in the training data should be the ratio we expect in testing data. Toward this end, we sample small windows of text from our corpus. The window length is sampled from a geometric distribution with a mean of 10 characters and length at least 3; these parameters are chosen based on our prior expectation of sign contents. In 10,000 samples, the same/different ratio is consistently about 0.057. This ratio controls the relative number of similar and dissimilar pairs we generated (100,000 total).

Our evaluation data comes from pictures of signs captured around a downtown North American city. There are 95 text regions (areas with the same font) and a total of 1,209 characters. Many signs have regular fonts (that is, characters appear the same in all instances) that are straightforward, such as basic sans serif, and should be easily recognized. Other signs contain regular fonts that are custom or rarely seen in the course of typical document recognition. Finally, there are a few signs with custom irregular fonts that pose the greatest challenge to the premise that similarity information is useful. The signs are imaged without extreme perspective distortion—they are roughly fronto-parallel. If this were not the case, affine rectification methods could be applied to the image [5]. Examples from the data can be seen in the sections that follow.

Since the focus of this work is recognition, we have annotated our evaluation data with the bounding boxes for characters. The character height is normalized and only filter responses from within the bounding box are considered when calculating the energies for character identity U^G and similarity U^S. Previous research has focused on doing this automatically, both in the context of signs [5] and OCR [4]. Note that Gabor filters are applied to the actual grayscale image; no binarization is performed.

3.2. Training Details

As mentioned previously, the ideal prediction is a result of integrating for a Bayesian posterior. Although recent advances have been made in approximation of the integral [15], the standard approach of using the mode of the model posterior (2) for prediction, rather than a weighted average of models, proves to give reasonable results in our experiments.

To avoid the need for performing inference on several chains with interdependent features during training, we use the piecewise method of Sutton and McCallum [16]. The training data is broken into graphs that have disjoint energy functions, and the parameters $\boldsymbol{\theta} = (\mathbf{w}, \mathbf{b}, \mathbf{l}, \mathbf{s})$ are subsequently optimized. This is especially advantageous for the bigram and case switch models (14), (15), and (16), which do not depend on an observed image. Thus, training instances may be collapsed into unique cases and weighted by their frequency. For example, the corpus $\mathcal{T}$ of 49 million characters contains nearly 780,000 occurrences of the bigram `th`. Rather than doing inference on the entire chain of text with the exact method, we need only do inference once in a two-node chain for `th` and count it 780,000 times.

The character image parameters $\mathbf{w}$ are trained on 200 fonts, and the remaining 800 fonts are used as a validation set. The value of hyperparameter α for the Laplacian

[1] GNU Image Manipulation Program `http://www.gimp.org`.

[2] `http://www.gutenberg.org`

Information	Accuracy	FNR	FPR	HR
I_G	84.04	11.42	0.51	91.07
$I_G I_S$	84.04	11.42	0.51	91.07
$I_G I_B$	87.92	9.14	0.53	93.81
$I_G I_C$	87.92	8.79	0.87	94.03
$I_G I_B I_C$	91.65	6.85	0.66	98.68
$I_G I_B I_C I_S$	93.22	5.45	0.14	99.26

Table 1. Recognition results (percentages) of the model with varying amounts of information. Overall character accuracy as well as the false negative (FNR), false positive (FPR), and hit rate (HR) for pairs (see text) are given.

prior that yields the highest likelihood on the validation set is used for optimizing the posterior for $\mathbf{w}$. The filter outputs are scaled to 32×32 for the character identity energy U^G. Although some information from the highest frequency filters is lost, this reduces the dimensionality of $\mathbf{w}$ by a factor of 16. The full-size filter outputs are used to calculate the angle and subsequent distance between images. The finer details are useful for these comparisons, and the dimensionality is not an issue.

3.3. Results

The results of character recognition with varying amounts of information are given in Table 1. Using information I_X means adding the energies U^X to the model. The maximum posterior marginal (MPM) labeling

$$\widehat{y_i} = \arg\max_{y \in A} p\left(y_i \mid \mathbf{x}, \widehat{\boldsymbol{\theta}}, I\right) \qquad (20)$$

is used for inference, as opposed to the more typical MAP labeling. MPM tends to give slightly higher accuracies than MAP on our data and task, but with the same relative performance between different amounts of information. The marginals (which are exact except when I_S is used) are given by belief propagation. Loopy belief propagation fails to converge on three difficult signs. Accuracy is the percentage of characters correctly identified (including case). To evaluate the ability of our model to recognize different instances of the same character in the same font, for intra-sign and intra-font characters we measure:

False negative rate: Percentage of character pairs that are the same but are given different labels.

False positive rate: Percentage of character pairs that are different but are given the same label.

Hit rate: Percentage of character pairs that are the same, given the same label, and correct (correctly labeled true positives).

Information	Accuracy	FNR	FPR	HR
I_S	-	22.67	0.25	-
$I_S \rightarrow I_G$	83.54	7.03	0.69	88.28
$I_S \rightarrow I_G I_B$	87.92	4.39	0.80	91.73
$I_S \rightarrow I_G I_C$	87.76	5.80	1.02	92.72
$I_S \rightarrow I_G I_B I_C$	91.40	3.69	0.88	97.26

Table 2. Results of clustering followed by recognition and voting.

All of the differences in accuracy for the unified model (Table 1) are statistically significant. (In all cases, significance is assessed by a paired, two-sided sign test on the accuracy per sign.) In particular, adding the similarity information I_S to $I_G I_B I_C$ improves accuracy with significance at the $p < 0.02$ level. While the reduction of false negatives is not significant with the addition of I_S, the false positives are cut by 79% ($p < 0.0005$).

For comparison, the method of clustering letters (within a sign and font) was followed by voting on the cluster label. To cluster letters, we maximize $p\left(\mathbf{y} \mid \mathbf{x}, \widehat{\mathbf{s}}, I_S\right)$ for $\mathbf{y}$ via simulated annealing, initialized at the prediction given by I_G (the strategy taken by Breuel [2]). A labeling based on additional information (e.g., I_G and I_B) is then produced by assigning the majority label to all members of a cluster (ties are broken by choosing the label whose members have the lowest average entropy for their posterior marginal, a strategy which slightly outperforms random tie breaking).

The differences between false negative rates (say, between $I_G I_B$ and $I_S \rightarrow I_G I_B$) are all significant ($p < 0.005$). There are two significant differences between false positive rates: $I_G I_B$ versus $I_S \rightarrow I_G I_B$ ($p < 0.005$) and $I_G I_B I_C I_S$ versus $I_S \rightarrow I_G I_B I_C$ ($p < 0.000005$). The differences in accuracy are not significant, save for that of $I_G I_B I_C I_S$ versus $I_S \rightarrow I_G I_B I_C$ ($p < 0.005$).

Another interesting comparison is provided by the likelihood ratio of the data under models with different amounts of information. Let $\mathcal{D} = \left\{\mathbf{y}^{(k)}, \mathbf{x}^{(k)}\right\}_k$ represent our evaluation label and image data. The geometric mean of the likelihood ratio between the language-informed and appearance-only models is

$$\left[\prod_{k=1}^{N} \frac{p\left(\mathbf{y}^{(k)} \mid \mathbf{x}^{(k)}, \widehat{\boldsymbol{\theta}}, I_G, I_B, I_C\right)}{p\left(\mathbf{y}^{(k)} \mid \mathbf{x}^{(k)}, \widehat{\boldsymbol{\theta}}, I_G\right)}\right]^{\frac{1}{N}} \approx 85.33. \qquad (21)$$

Adding the similarity information to the model also yields an increase in belief about the correct labels for the data:

$$\left[\prod_{k=1}^{N} \frac{p\left(\mathbf{y}^{(k)} \mid \mathbf{x}^{(k)}, \widehat{\boldsymbol{\theta}}, I_G, I_B, I_C, I_S\right)}{p\left(\mathbf{y}^{(k)} \mid \mathbf{x}^{(k)}, \widehat{\boldsymbol{\theta}}, I_G, I_B, I_C\right)}\right]^{\frac{1}{N}} \approx 1.02. \qquad (22)$$

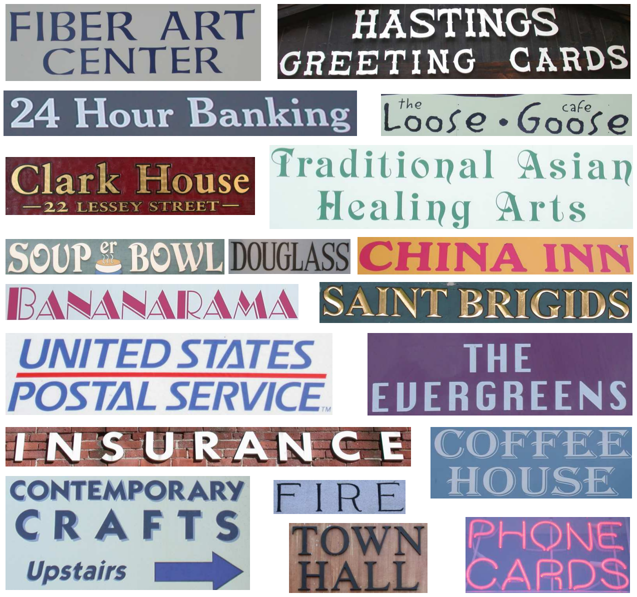

Figure 5. Examples of signs read correctly.

3.4. Discussion

Figure 5 contains examples of signs correctly read, showing that the features are robust to various fonts and background textures (e.g., wood and brick).

Although the number of characters per query is small compared to OCR applications, adding similarity information undoubtedly improves recognition accuracy, reducing overall error by nearly 20%. Not surprisingly, most of this improvement comes from greatly reducing the cases when different characters are given the same label (pair false positives).

Perhaps surprisingly, adding similarity information to the simple image information I_G does not alter the results. This is probably because test images have relatively little noise and are mostly difficult due to font novelty and nonfronto-parallel orientations. Therefore, it is expected that the same characters, though novel, would often be given the same label in different locations, due to their logical independence solely with information I_G. However, when other sources of information are introduced to help resolve ambiguity, the similarity information does make a difference because the bigram and case information are based on local context. This can push the beliefs about characters in different directions, even though they tend to look the same, because their contexts are different. Adding the similarity information on top of these other sources ensures that the local context does not introduce a contradictory bias. This is demonstrated in Figure 1. Adding bigram information pushes the second e to an a because preference for the ea bigram outweighs both ee and the character/image energy.

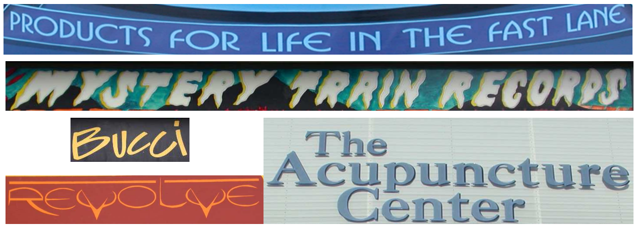

Figure 6. Challenging signs that have unique fonts, are hand-painted, or contain three-dimensional effects, real and virtual.

Similarly, adding case information pushes the l from being recognized as the upper case I to lower case t (due to kerning in this italic font, some of the F overlaps in the l's bounding box, leaving a little crossbar indicative of a t). Finally, adding the similarity information corrects the l since it is very different from the final t, and corrects the es since they are very similar.

As expected, adding more prior information to the model boosts the likelihood of the data. The model using appearance alone is relatively weak, since a probability has an upper bound of one, yet the ratio in (21) is quite large. In addition to improving the prediction accuracy, adding the similarity information yields an increase in the degree of belief for the correct labels, as shown by (22). Although the increase is slight on average, more than ten percent of the signs in our test data exhibit an increase of at least one order magnitude. This could be important when confidence in the model's prediction helps to determine how to handle a query.

The results of clustering the letters prior to recognition appear worse than doing recognition outright with no similarity information, though the difference is not significant. However, unifying all the information available does yield better results than a distinct clustering step. It is interesting that clustering yields fewer false negatives than the unified approach. This is most likely because clusters are not forced to have different labels at the secondary assignment stage. Thus, instances of the same character assigned to different clusters are not forced to have different labels (up to the fact that there are only as many clusters as characters in our alphabet A). Indeed, if this *were* the case, the false negative rate would be intolerably high. Conversely, the clustering preprocessing step does commit unrecoverable errors by pairing characters that are not the same; subsequent information cannot reduce the false positive rate. This is especially critical because the probability of two characters being the same *a priori* is much smaller than their being different, thus the false positive rate has a greater impact on total errors than the false negative rate.

Some signs in our data set present tremendous difficulty and challenge the assumption that characters of the same "font" appear similar. Some of these are due to rendered warping effects, custom fonts, or inconsistent shadow effects (see Figure 6). Other signs just have unique fonts that are very different from those in the training set.

4. Conclusions

There are many important sources of information that allow humans to easily solve recognition tasks that remain challenging for computers. Object similarity is an underused source of information in problems involving novel input sources. In this paper, we have shown that using similarity information can improve overall accuracy and greatly reduce the particular error of giving different items the same label. Our results show that the benefit of additional information is maximized when used alongside all other information within a unified framework, rather than in distinct stages that make error recovery impossible. The probabilistic model allows the various interpretations to be weighed against each other in light of all available information before a final prediction must be made.

While we have demonstrated the utility of a unified approach to similarity on an application reading signs in natural scenes, it may prove useful in other tasks such as speech and generic 3D object recognition.

Acknowledgements

This work was supported in part by NFS grant IIS-0100851. Thanks to Piyanuch Silapachote and Marwan A. Mattar for taking most of the sign pictures, Charles Sutton for many helpful discussions, and Andrew Fast for comments on this manuscript.

References

[1] A. Brakensiek, D. Willett, and G. Rigoll. Improved degraded document recognition with hybrid modeling techniques and character n-grams. In *Proc. Intl. Conf. on Pattern Recognition*, volume 4, pages 438–441, 2000.

[2] T. M. Breuel. Classification by probabilistic clustering. In *Proc. Intl. Conf. on Acoustics, Speech, and Signal Processing*, volume 2, pages 1333–1336, 2001.

[3] T. M. Breuel. Character recognition by adaptive statistical similarity. In *Proc. Intl. Conf. on Document Analysis and Recognition*, volume 1, pages 158–162, 2003.

[4] R. G. Casey and E. Lecolinet. Strategies in character segmentation: a survey. In *Proc. Intl. Conf. on Document Analysis and Recognition*, volume 2, page 1028, Washington, DC, USA, 1995.

[5] X. Chen, J. Yang, J. Zhang, and A. Waibel. Automatic detection and recognition of signs from natural scenes. *IEEE Transactions on Image Processing*, 13(1):87–99, 2004.

[6] X. Chen and A. L. Yuille. Detecting and reading text in natural scenes. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 366–373, 2004.

[7] J. G. Daugman. Complete discrete 2-D Gabor transforms by neural networks for image analysis and compression. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 36(7):1169–1179, 1988.

[8] D. Deng, K. Chan, and Y. Yu. Handwritten Chinese character recognition using spatial Gabor filters and self-organizing feature maps. In *Proc. Intl. Conf. on Image Processing*, volume 3, pages 940–944, 1994.

[9] J. Goodman. Exponential priors for maximum entropy models. Technical report, Microsoft Research, One Microsoft Way, Redmond, WA 98052-6399, 2003.

[10] J. D. Hobby and T. K. Ho. Enhancing degraded document images via bitmap clustering and averaging. In *Proc. Intl. Conf. on Document Analysis and Recognition*, volume 1, pages 394–400, 1997.

[11] T. Hong and J. J. Hull. Improving OCR performance with word image equivalence. In *Symposium on Document Analysis and Information Retrieval*, pages 177–190, 1995.

[12] F. Kschischang, B. Frey, and H.-A.Loeliger. Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, 47(2):498–519, Feb. 2001.

[13] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. 18th International Conference on Machine Learning*, pages 282–289. Morgan Kaufmann, San Francisco, CA, 2001.

[14] B. Manjunath and W. Ma. Texture features for browsing and retrieval of data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(9):837–842, 1996.

[15] Y. Qi, M. Szummer, and T. P. Minka. Bayesian conditional random fields. In *Proc. 10th Intl. Workshop on Artificial Intelligence and Statistics (AISTATS05)*, 2005.

[16] C. Sutton and A. McCallum. Piecewise training of undirected models. In *21st Conference on Uncertainty in Artificial Intelligence*, 2005.

[17] P. M. Williams. Bayesian regularization and pruning using a Laplace prior. *Neural Computation*, 7:117–143, 1995.

[18] V. Wu, R. Manmatha, and E. M. Riseman. Finding text in images. In *DL'97: Proceedings of the 2nd ACM International Conference on Digital Libraries*, pages 3–12, 1997.